DAta Mining & Exploration Program

Dipartimento di Scienze Fisiche
Università di Napoli "Federico II"

ISTITUTO NAZIONALE di ASTROFISICA
OSSERVATORIO ASTRONOMICO di CAPODIMONTE

CALTECH

DAME Web Application REsource

Plugin Setup Tool

User Manual

DAMEWARE-MAN-NA-0016

Issue: 1.0
Date: October 15, 2011
Authors: M. Brescia, S. Riccardi

Doc. : ModelPlugin_UserManual_DAMEWARE-MAN-NA-0016-Rel1.0

INAF-OACN

IVOA

This document contains proprietary information of DAME project Board. All Rights Reserved.

Index

1 Introduction 4
2 System Requirements and Installation 4
3 The Java plugin wizard GUI 5
3.1 General Plugin Information 6
3.2 Use Case Specifications 7
3.3 Input parameter Setup 8
3.3.1 New parameter setup 9
3.3.2 Edit, sort and remove input parameter setup 11
3.4 Input file Setup 12
3.4.1 Add new input files 12
3.4.2 Edit, sort and remove an input file 13
3.5 Output File Setup 14
3.5.1 Add new output files 14
3.5.2 Edit and remove an output file 16
3.6 Command line constant parameters Setup 17
3.6.1 Add a constant field 17
3.6.2 Edit, sort and remove a constant field 18
3.7 The Full use case 18
3.8 Final Steps of the setup procedure 20
3.9 Delivery of the plugin setup tool 21
4 Appendix – References and Acronyms 22

TABLE INDEX

Tab. 1 – Abbreviations and acronyms *22*
Tab. 2 – Applicable Documents *23*

FIGURE INDEX

Fig. 1 – introduction panel of the plugin wizard tool after launch *5*
Fig. 2 – main procedure buttons *5*
Fig. 3 – loading XML plugin configuration file panel *6*
Fig. 4 – General model information panel *6*
Fig. 5 – Error message (spaces are not allowed) *7*
Fig. 6 – functionalities to be selected for the new model *7*
Fig. 7 – use case setup panel *8*
Fig. 8 – General panel to build new parameter fields (Train use case example) *9*
Fig. 9 - Example of error of assign incorrect *10*
Fig. 10 – Example (Train use case) panel for new parameter (field) editing *10*
Fig. 11 – Data types *10*
Fig. 12 – selection of field value constraints *10*
Fig. 13 – Example (Train use case) of input parameter editing panel *11*
Fig. 14 – Example (Train use case) of input file setup panel *12*
Fig. 15 – Example (Train use case) of dataset input file adding panel *12*
Fig. 16 – available format types for dataset files supported by the model to be plugged in *13*
Fig. 17 – Example (Train use case) of input file editing panel *13*
Fig. 18 – Example (Train use case) of the output file setup panel *14*
Fig. 19 – Example (Train use case) of output file adding setup panel *14*
Fig. 20 – scatter plot setup window *15*
Fig. 21 – input file type selection to be joined in the generated output file *15*
Fig. 22 – Example (Train use case) of output file editing panel *16*
Fig. 23 – Example (Train use case) of "additional fields and sorting" setup panel *17*
Fig. 24 – window to add a constant field *17*
Fig. 25 – types selectable for data fields *17*
Fig. 26 – Example of constant editing *18*

This document contains proprietary information of DAME project Board. All Rights Reserved.

Fig. 27 – panel for Full use case *19*
Fig. 28 – specify type of the file movable from experiment to dataset area of the web app GUI *19*
Fig. 29 – final panel..... *20*
Fig. 30 – select directory where to save created files..... *20*
Fig. 31 – example of message notifying the successful creation of the compressed plugin package..... *21*

This document contains proprietary information of DAME project Board. All Rights Reserved.

1 Introduction

The present document is the user guide of the DAMEWARE plugin setup tool, created to make able end users of the data mining web app to easily integrate new machine learning models within the DAMEWARE infrastructure, without any specific knowledge about its internal architecture.

This manual, together with the plugin Java program, is part of the web app beta release documentation package, and it is available at the address:

http://dame.dsf.unina.it/beta_info.html,

where there is also the direct gateway to the beta web app.

The plugin setup tool, in the current version 1.0, through a guided procedure, allows user to configure all machine learning model parameters, use cases and I/O information setup. Then it generates source code to be physically included into the infrastructure and to be finally exposed as new machine learning resource for the community.

The plugin setup tool implicitly assumes that the user model to be plugged in is available in the form of an executable, provided with a command line parameter list[1] or with external configuration files.

2 System Requirements and Installation

The only basic requirement to download and execute the Java plugin setup tool on local user machine is that the Java Runtime Environment (JRE) is properly installed. If not, please download and install the JRE from

http://www.oracle.com/technetwork/java/javase/downloads/index.html

After JRE installation, download the plugin application from the web address specified in the introduction above. Then simply right click on the .jar file and select “Open with” and “Java Platform SE binary”.

Alternatively, you can launch the .jar application directly from the system window, by executing at prompt the following command line:

> **java –jar DAMEWAREPlugin.jar [return]**

[1] In the next releases it will also possible to use external configuration files instead of command line parameters

This document contains proprietary information of DAME project Board. All Rights Reserved.

3 The Java plugin wizard GUI

The following sections will drive users through the plugin integration procedure.
In Fig. 1 the intro panel is shown. It appears after the execution of the wizard application.

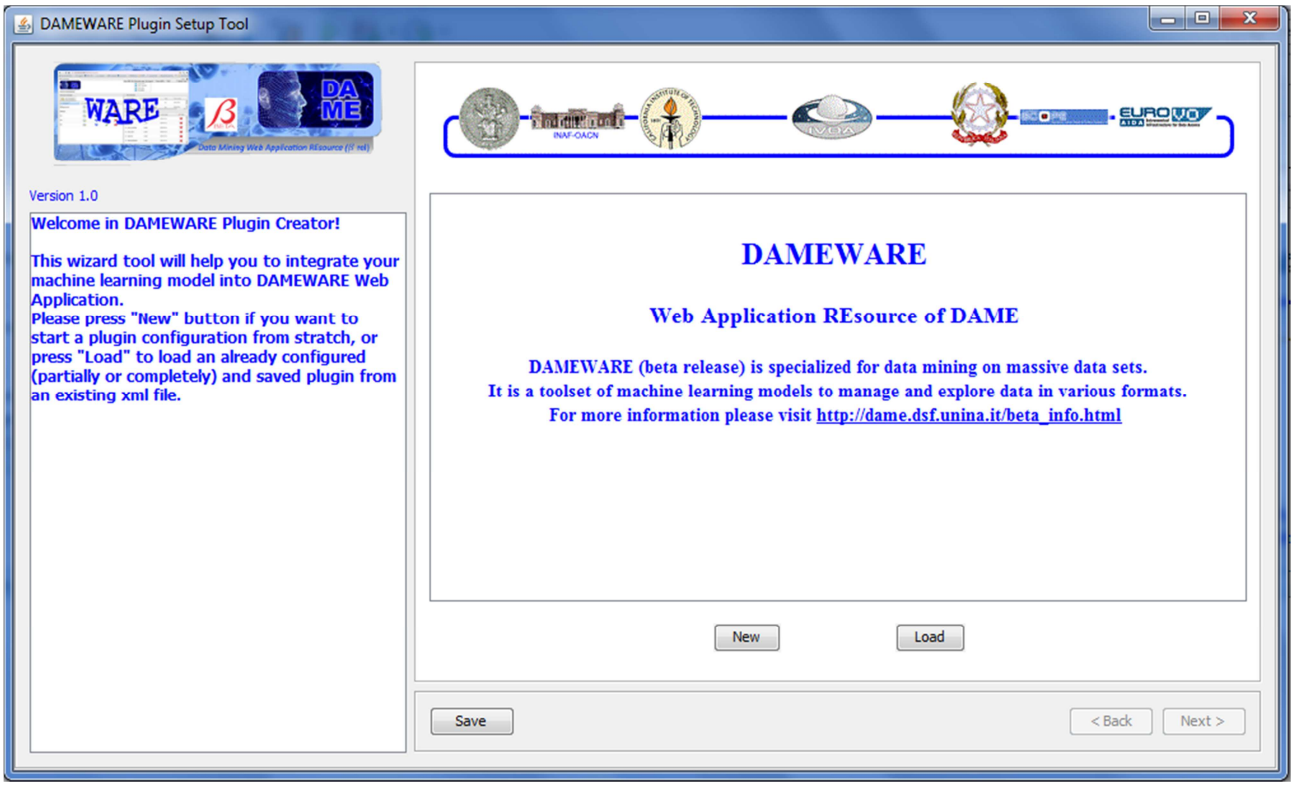
DAMEWARE Plugin Setup Tool
Version 1.0
Welcome in DAMEWARE Plugin Creator!
This wizard tool will help you to integrate your machine learning model into DAMEWARE Web Application.
Please press "New" button if you want to start a plugin configuration from stratch, or press "Load" to load an already configured (partially or completely) and saved plugin from an existing xml file.
DAMEWARE
Web Application REsource of DAME
DAMEWARE (beta release) is specialized for data mining on massive data sets.
It is a toolset of machine learning models to manage and explore data in various formats.
For more information please visit http://dame.dsf.unina.it/beta_info.html
New
Load
Save
< Back
Next >

Fig. 1 – introduction panel of the plugin wizard tool after launch

At the bottom of the intro panel there are two buttons “Back” and “Next” (Fig. 2), initially disabled, useful to proceed (forward and backward respectively) through the procedure steps, described in the next sections.

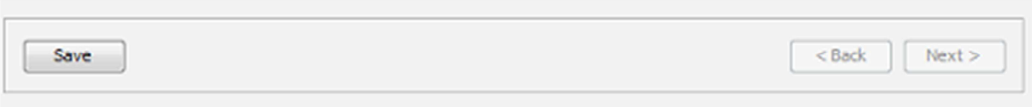

Fig. 2 – main procedure buttons

The button “New” is used if user wants to start a plugin configuration from scratch. Otherwise it can be selected the “Load” button to load an already configured (partially or completely) and saved plugin.
In this last case, after selecting the “Load” option a new window will let you choose any of the XML configuration files available (Fig. 3).

This document contains proprietary information of DAME project Board. All Rights Reserved.

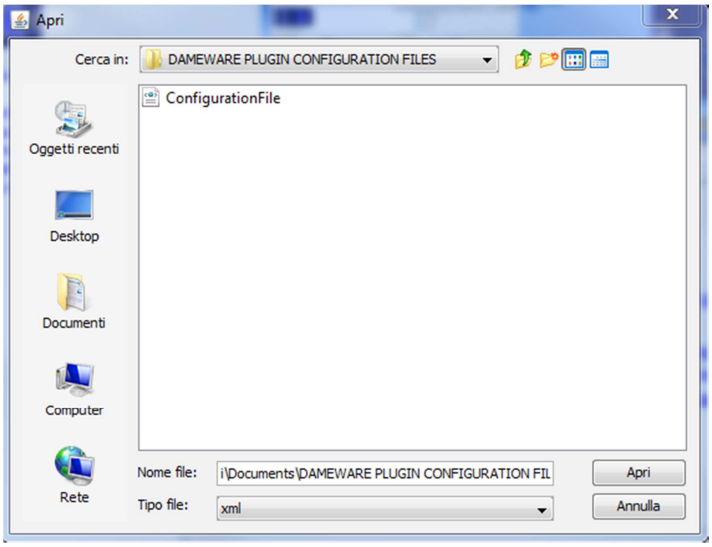

Fig. 3 – loading XML plugin configuration file panel

In the following we will focus on the "New" option, starting the new plugin configuration procedure.

3.1 General Plugin Information

The Fig. 4 shows the general plugin information to be compiled.

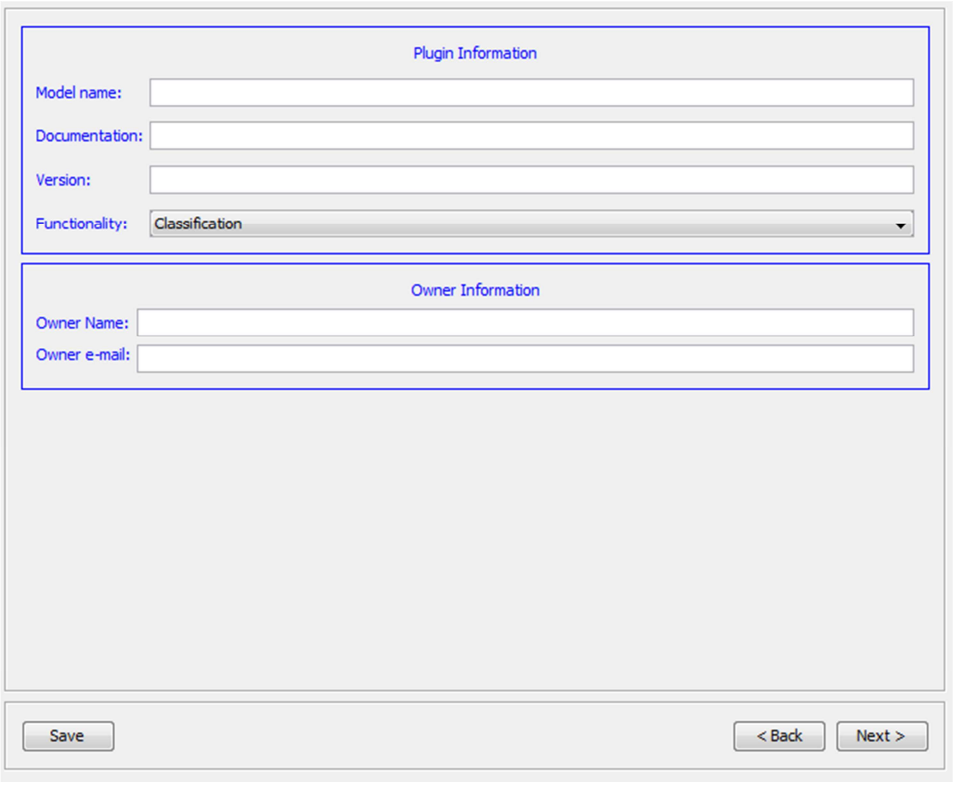

Fig. 4 – General model information panel

This document contains proprietary information of DAME project Board. All Rights Reserved.

- **Model name**: the name of the model to be plugged in. It is strongly suggested to use an acronym of the model (for example MLP for Multi Layer Perceptron). Spaces are not allowed here (Fig. 5), use underscore instead, if required;
- **Documentation**: the URI address where to find model related documentation (if absent, any reference can be specified as well);
- **Version**: current model version or release;
- **Functionality**: the functional domain of experiments to be associated to the plugged model (for example classification or regression for supervised models, clustering for unsupervised etc…). The user should be aware about the feasible domain for the model (Fig. 6);
- **Owner Name**: the user name;
- **Owner Mail**: a correct e-mail address of the user, to be contacted by DAMEWARE administrator;

At the end, click "Next" button to proceed.

Fig. 5 – Error message (spaces are not allowed)

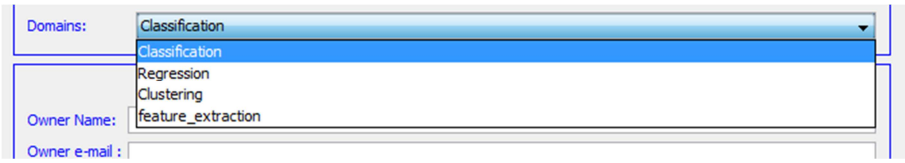

Fig. 6 – functionalities to be selected for the new model

3.2 Use Case Specifications

As common practice in the machine learning domain, any experiment requires a series of phases (use cases) to be performed in sequence. For instance, at the beginning of any experiment, a training phase is required to train the model (either in supervised and unsupervised cases). The test phase is usually the second step (sometimes also named as validation & test phase). The last step is the "run" phase, in which the trained and tested model is used as a generic function, applied to new data of the focused problem. There is also the "full" step, automatically performing both "train" and "test" phases, considered as a unique workflow.

This document contains proprietary information of DAME project Board. All Rights Reserved.

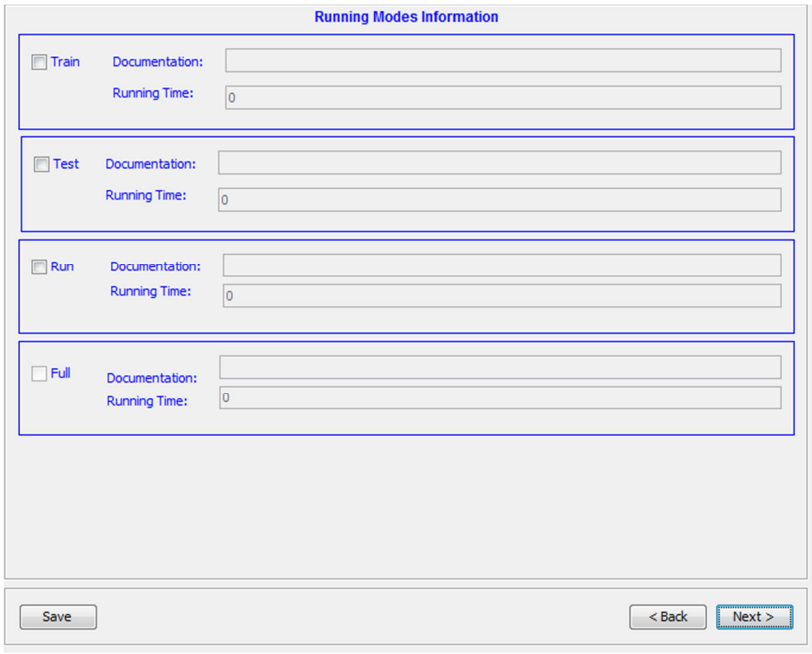

Fig. 7 – use case setup panel

Fig. 7 shows the use case setup panel, in which the user can choose the steps available for the plugged model (usually a canonical machine learning includes all of them), by also specifying referenced documentation (if any) and running time[2] priorities (usually the user can leave default values if not aware).

If train and test cases checkboxes are selected, the "full" case is automatically included, otherwise it is recommended to unselect "full" case if train or test case is not chosen.

3.3 Input parameter Setup

In Fig. 8 the main input data panel is shown. It allows user to enter information about the parameters required by the model for each of the previously selected use cases. The figure reports an example for the Train use case (it is the same in other use cases).

[2] Running time fields functionality will be available in the next releases

This document contains proprietary information of DAME project Board. All Rights Reserved.

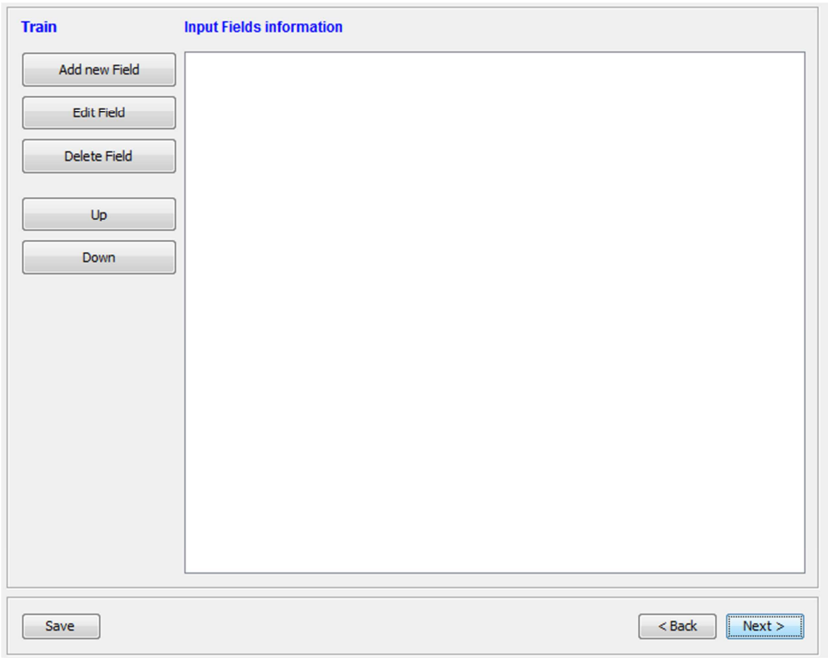

Fig. 8 – General panel to build new parameter fields (Train use case example)

3.3.1 New parameter setup

By selecting "Add new field" in the panel of Fig. 8, the panel in Fig. 10 allows to configure the input parameter by specifying:

- **Visualized label**: the label of the parameter as it will appear in the DAMEWARE web app experiment setup tab (for example "input layer neurons" or "error threshold");
- **Description**: a brief description of the parameter;
- Type: the internal value allowed for the parameter (integer, double, string), as listed in Fig. 11;
- **Is Optional**: checkbox to be selected if the parameter will be not required (for an optional parameter, default value will be assigned if left empty);
- **Value**: if the "is optional" checkbox is selected, the default value field will become enabled and the user will have to enter the desired value (according to the type specified in the field "Type"); If desired value is not according to the type specified in the field "Type" an Error message will be shown (Fig. 9);
- **Ucd**: for uniformity with VOTABLE standards, this field can be used to specify the "Uniform Content Descriptor" of the parameter[3] (if any). The user not aware can leave empty this field;
- **Unit**: optionally the user can specify the measurement unit of the parameter. This attribute specifies the units in which the values of the corresponding column are expressed. The quantities in a column of the table may be expressed in some physical unit, which is specified by the unit attribute of the FIELD. It basically consists as a string without blanks or spaces, where the symbol "." or "*" indicates a multiplication, "/" stands for the division, and no special symbol is required for a power. Examples are unit="m2" for m^2, unit="cm-2.s-1.keV-1" for $cm^{-2}s^{-1}keV^{-1}$, or unit="erg/s" for erg s^{-1};
- **Precision**: optionally the user can enter the floating point precision for the parameter internal representation (e.g. precision="F2" or equivalently precision="2" to express 2 significant digits after the decimal point), or as a number of significant digits (e.g. precision="E5" indicates a relative precision of 10^{-5}). Examples: precision="4" unit="deg"; precision="1" unit="s";
- **Utype**: it is a not mandatory attribute of any RESOURCE TABLE FIELD. It gives a fully detailed meaning of the field, parameter or group.

[3] http://www.euro-vo.org/pub/tc/tutorials/tutorial-3.html

This document contains proprietary information of DAME project Board. All Rights Reserved.

- (Example: Utype=”stc:AstroCoords.Position2D.Value2.C1”)

- **Constraint**: if the parameter has a limited range of allowed values, the user can choose one of the options reported in Fig. 12. Choose NO_CONSTRAINT otherwise;
- **Constraint Values**: if VALUES option has been selected in the previous field, here the list of allowed values must be entered (values separated by commas, example 0,1,2). If the previous choice was RANGE, here the interval range must be entered (by using : as mid character, example 0.1:0.9).

Fig. 9 - Example of error of assign incorrect

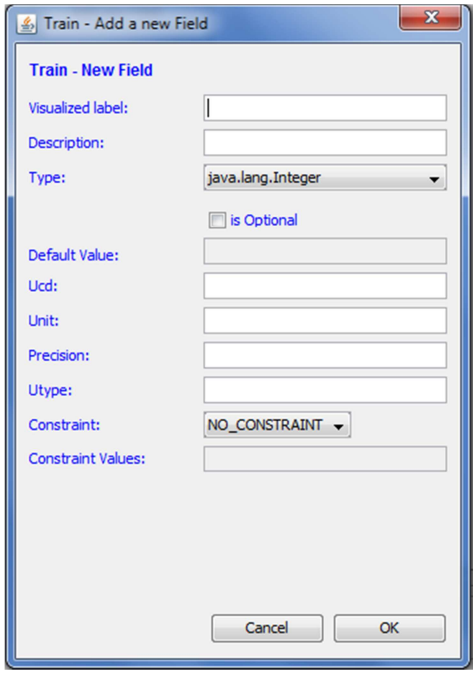

Fig. 10 – Example (Train use case) panel for new parameter (field) editing

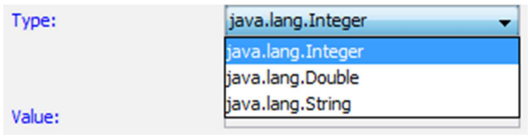

Fig. 11 – Data types

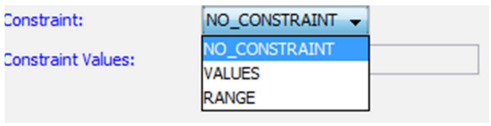

Fig. 12 – selection of field value constraints

At the end, press “OK” button to save the parameter configuration.

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.3.2 Edit, sort and remove input parameter setup

By selecting “Edit Field” button in the panel of Fig. 8, it is possible to modify the saved parameter configuration (Fig. 13).

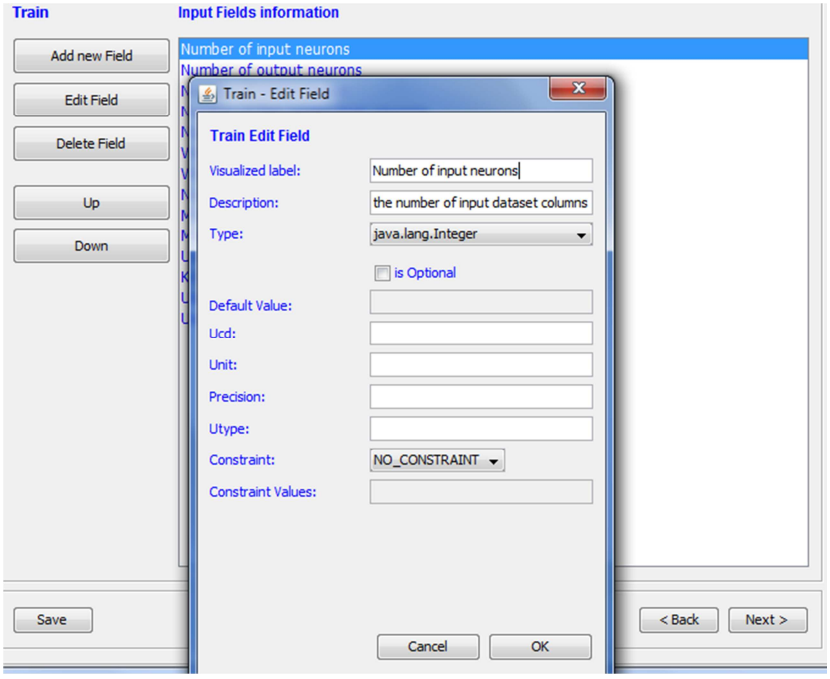

Fig. 13 – Example (Train use case) of input parameter editing panel

By selecting “Delete Field” button in the panel of Fig. 8, it is possible to delete the saved configuration for a parameter. Finally, the buttons “Up” and “Down” in the panel of Fig. 8 can be used to sort listed parameters (already configured) as they will appear in the sequence shown in the DAMEWARE web app model configuration tab during an experiment setup process.

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.4 Input file Setup

Any use case of a model foresees of course to specify input files to be used by the model (such as dataset, trained weight matrix for a neural network, external configuration files etc.).
The panel shown in Fig. 14 reports an example of input file setup widget (for instance in the Train use case, the same for the other use cases).

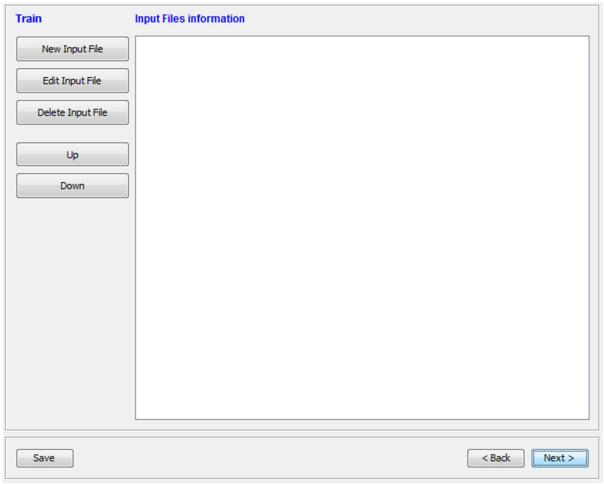

Fig. 14 – Example (Train use case) of input file setup panel

3.4.1 Add new input files

By selecting "New Input File" in the panel of Fig. 14, the panel of Fig. 15 appears, allowing the user to enter the file information.

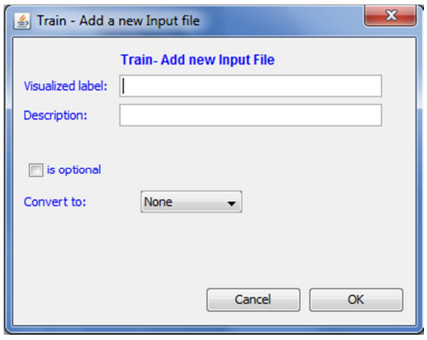

Fig. 15 – Example (Train use case) of dataset input file adding panel

- **Visualized label**: the name of the input file type label (as it will appear in the DAMEWARE web app setup tab). For example: "dataset" or "trained weight file
- **Description**: a brief description of the file content;
- **Is optional**: if selected this input file field is considered not required (if optional it is empty, no default values will be used in this case);
- **Convert to**: if the model requires a particular format for this file (for example, the dataset should be given as a CSV file), the user can choose the right format by selecting one of the allowed formats listed in the pop-up menu shown in Fig. 16. **NOTE**: the DAMEWARE web app allows to arbitrarily submit input files in a quite wide range of supported formats. But the model usually allows only a particular format as input. So far, the web app will perform an embedded conversion to the right format, as specified by the user in this field.

This document contains proprietary information of DAME project Board. All Rights Reserved.

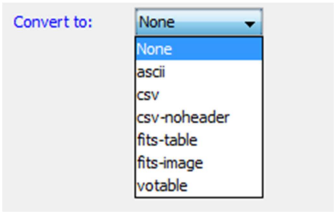

Fig. 16 – available format types for dataset files supported by the model to be plugged in

At the end, click "OK" button to save the input file configuration.

3.4.2 Edit, sort and remove an input file

By selecting "Edit Input File" button in the panel of Fig. 14, it is possible to modify the saved file configuration (Fig. 17).

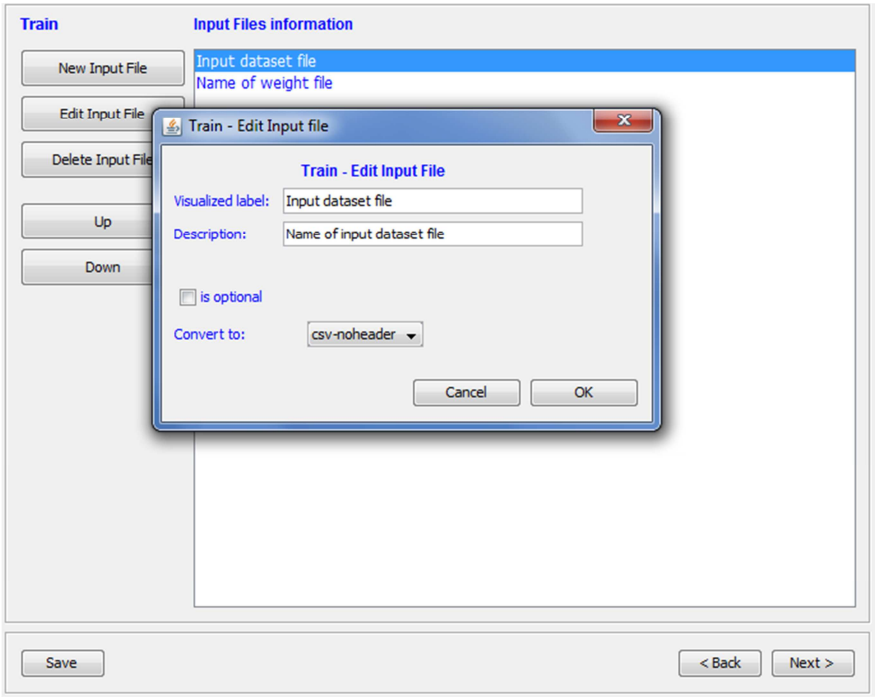

Fig. 17 – Example (Train use case) of input file editing panel

By selecting "Delete Input File" button in the panel of Fig. 14, it is possible to delete the saved configuration for an input file.
Finally, the buttons "Up" and "Down" in the panel of Fig. 14 can be used to sort listed input files (already configured) as they will appear in the sequence shown in the DAMEWARE web app model configuration tab during an experiment setup process.

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.5 Output File Setup

Each time an experiment is completed, involving one or more use cases, its outcome consists in a series of output files, images, plots, diagrams, tables, trained weights, logs etc.
Through the panel shown in Fig. 18, the user can configure and insert the list of output files as foreseen to be produced for each model use case. The figure reports an example in the Train use case (the same for the others). Only the files specified here will be exposed as output available to be downloaded in the DAMEWARE web app experiment status window.

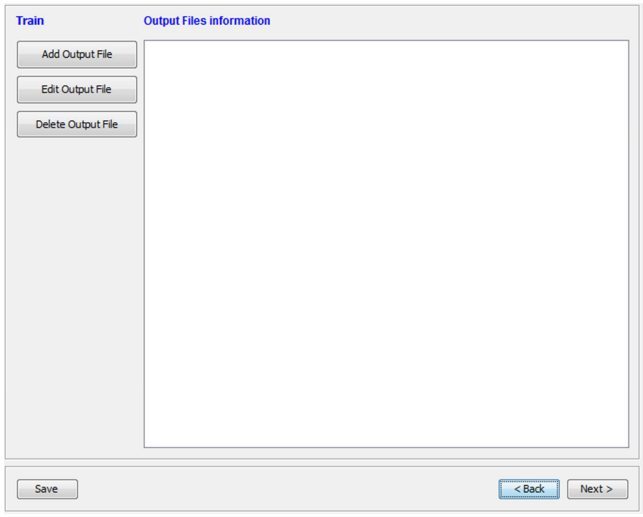

Fig. 18 – Example (Train use case) of the output file setup panel

3.5.1 Add new output files

By selecting "Add Output File" in the panel of Fig. 18, the panel of Fig. 19 allows the user to describe the file information.

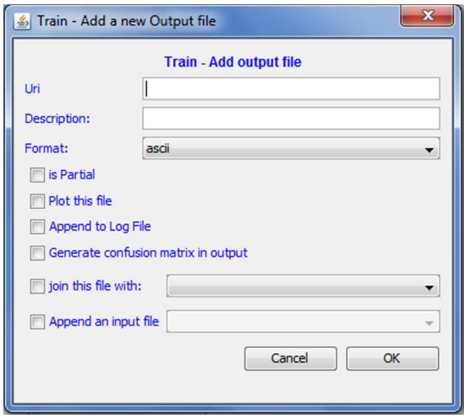

Fig. 19 – Example (Train use case) of output file adding setup panel

- **Uri**: relative path (sub-directory) where output file will be created (basically the experiment location used internally by the DAMEWARE web app);
- **Description**: a brief description of the file content;
- **Format**: the internal representation of the file (it is basically to assign a right extension at the creation time for further use in the web app);

This document contains proprietary information of DAME project Board. All Rights Reserved.

- **Is Partial**: selected only if the file will contain partial information (registered at runtime during experiment execution. For example a training error report file);
- **Plot this file**: select this checkbox only if it is required a plot for the file contents. If selected a pop-up tab will appear asking to configure plot features (Fig. 21):
  - **X**: file column index[4] to be used as the abscissa;
  - **Y**: file column index to be used as the ordinate;
  - **Name**: this field specifies the name of the parameter used as identifier for the produced image. (Example: X = 1, Y = 2, name = plotting1; Then the internal identifier of the image will be named plottingImage1.
  - **List of images to create**: a field where to insert the list of all the images to be generated from a specified output file. describing the images (Fig. 22 for an example);
- **Append to Log File**: it specify if the file content should also be included in the log file;
- **Generate confusion matrix in output**: the output values of the experiment (table of data) will be also saved as a confusion matrix (only in case of classification type experiments);
- **Join this file with**: Join it with columns from another output file;
- **Append an input file**: if selected the user must enter the type of the input file that will be joined into the output file (Fig. 22). The resulting output file will contain joined input and output columns.

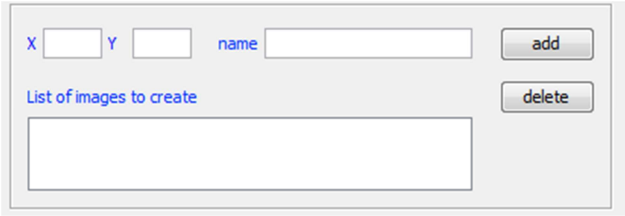

Fig. 20 – scatter plot setup window

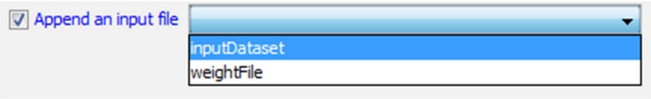

Fig. 21 – input file type selection to be joined in the generated output file

[4] Here index is the ordinal number referred to the column position in the file, not the meta-datum.

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.5.2 Edit and remove an output file

By selecting “Edit Output File” button in the panel of Fig. 18, it is possible to modify the saved file configuration (Fig. 22).

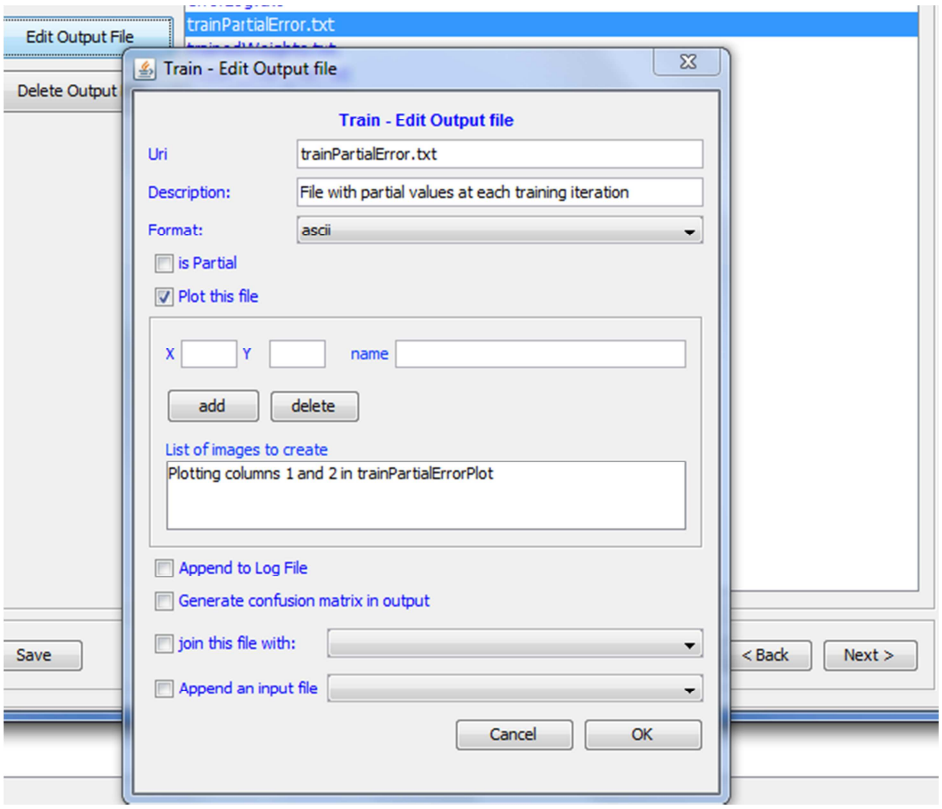

Fig. 22 – Example (Train use case) of output file editing panel

By selecting “Delete Output File” button in the panel of Fig. 18, it is possible to delete the saved configuration for an output file.

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.6 Command line constant parameters Setup

The “Additional fields and sorting” panel, shown in Fig. 23 (as example for the Train use case) allows user to specify additional field information in terms of command line constant parameters, as requested by the model executable to plugged in.

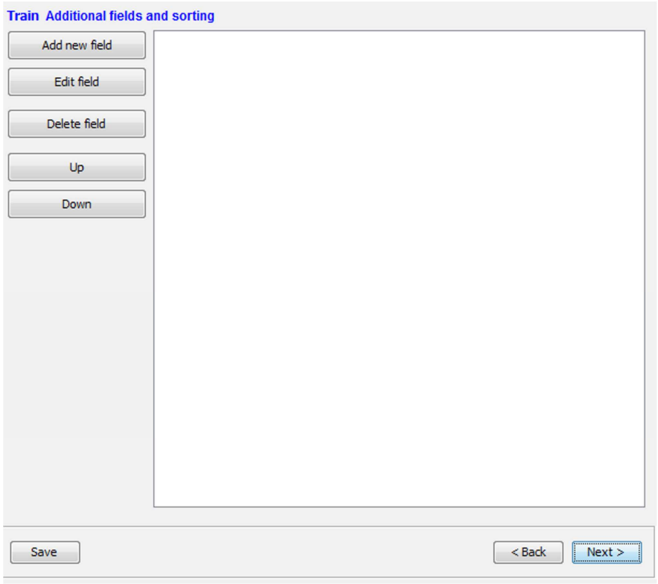

Fig. 23 – Example (Train use case) of “additional fields and sorting” setup panel

3.6.1 Add a constant field

By selecting “Add new field” in the panel of Fig. 23 it is possible to add a new command line constant parameter (Fig. 24). This can be done by describing its name (identifier of the parameter), type (Fig. 25) and numerical value.

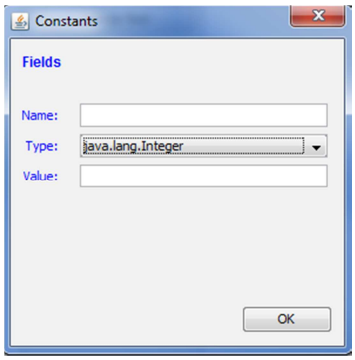

Fig. 24 – window to add a constant field

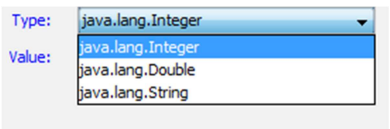

Fig. 25 – types selectable for data fields

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.6.2 Edit, sort and remove a constant field

After creation of a new constant command line parameter, it can be modified by choosing “Edit field” in the panel of Fig. 23).

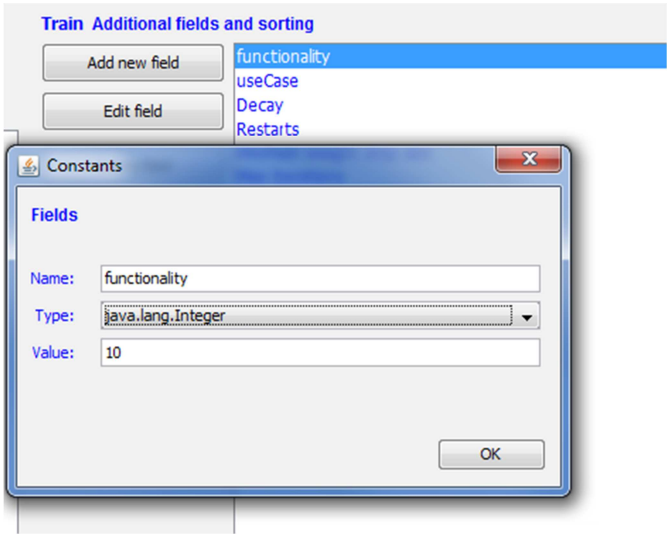

Fig. 26 – Example of constant editing

Of course, in a command line it is important the order in which the parameters are listed (otherwise the execution will crash). So far, to correctly locate the constant parameter position in the command line, the buttons “Up and “Down” of Fig. 23 can be used.
Finally the option “Delete field” in Fig. 23 allows user to delete a constant parameter.

3.7 The Full use case

As underlined before, the Full use case is a particular phase of an experiment, in which the Train and Test cases are performed together in sequence. If the Full case is selected the panel in Fig. 27 allows the user to specify the interface step information between the end of Train and the beginning of the Test cases.
As common in machine learning experiments, some outputs of Training phase become inputs for the Test phase (for instance in a neural network, the trained weights stored as an output file must be passed as input to the Test experiment). The panel in Fig. 27 is useful to specify the list and types (Fig. 28) of training output files that must be used as input files for the Test phase.

This document contains proprietary information of DAME project Board. All Rights Reserved.

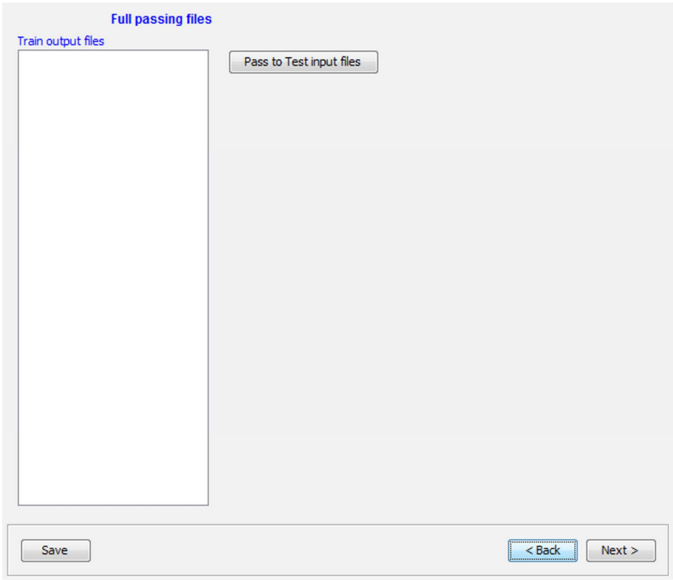

Fig. 27 – panel for Full use case

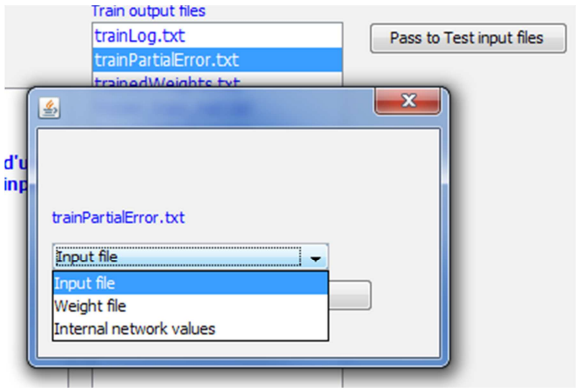

Fig. 28 – specify type of the file movable from experiment to dataset area of the web app GUI

This document contains proprietary information of DAME project Board. All Rights Reserved.

3.8 Final Steps of the setup procedure

At the end of the setup procedure, by having configured input and output files, constant and parameters of command lines for all the foreseen use cases of the model to be plugged in, the panel shown in Fig. 29 asks for the final name of the executable program to be launched if the plugin will be selected for an experiment through the DAMEWARE web app GUI.

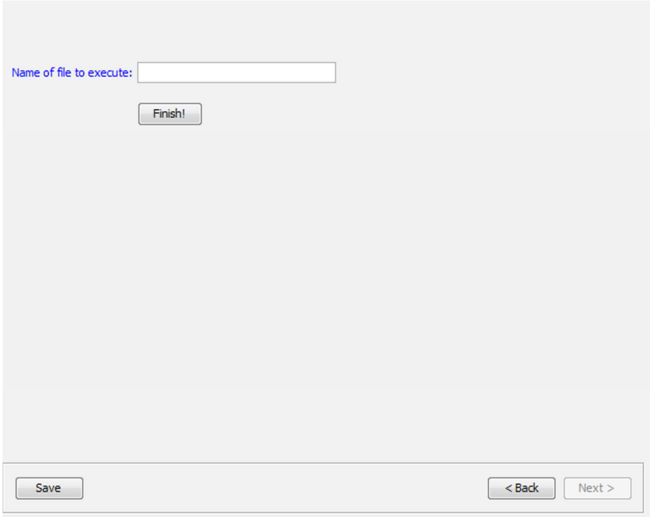

Fig. 29 – final panel

After the “Finish” button is clicked the procedure will create all internal features to be physically integrated into the DAMEWARE web app infrastructure.
The “Finish” button will activate a new panel, shown in Fig. 30, asking for the directory where to store all created files of the plugin package.

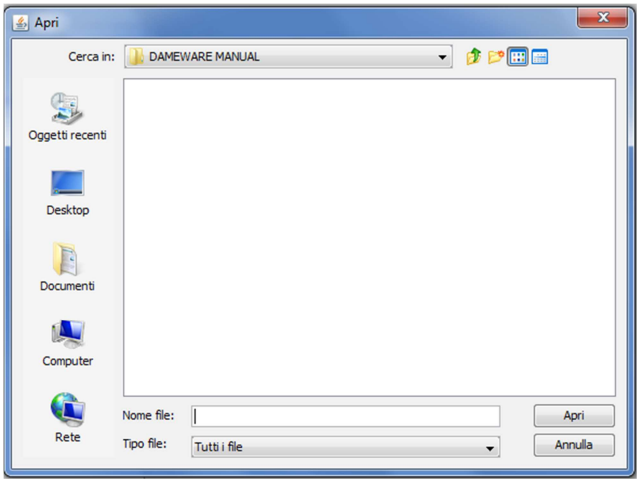

Fig. 30 – select directory where to save created files

The Fig. 31 reports the disclaimer which informs about the compressed archive successful creation.

This document contains proprietary information of DAME project Board. All Rights Reserved.

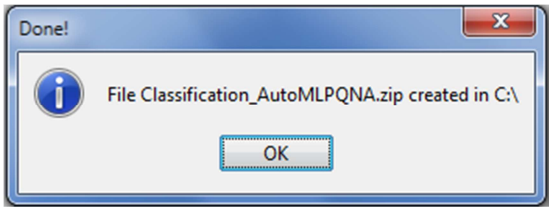

Fig. 31 – example of message notifying the successful creation of the compressed plugin package

3.9 Delivery of the plugin setup tool

At the end of the guided procedure, for instance when the user selects the button "Finish!" of Fig. 29, the setup tool creates a compressed package (.zip) called <functionality>_<model acronym>.zip, containing:

- ✓ A Java file named <functionality>_<model acronym>.java;
- ✓ A Java file named <model acronym>.java;
- ✓ An XML file named <functionality>_<model acronym>.xml;
- ✓ An XML file named ConfigurationFile.xml that tracks all the setup work done.

The user must deliver the .zip file, together with the source code of the model (to be recompiled on the worker nodes of DAME computing platform), to the DAMEWARE administrators (helpdame@gmail.com), asking to complete the model plugin integration task.
Then the user will be informed about the availability of the new model in the new release of DAMEWARE web app.

This document contains proprietary information of DAME project Board. All Rights Reserved.

4 Appendix – References and Acronyms

Abbreviations & Acronyms

A & A	Meaning	A & A	Meaning
AI	Artificial Intelligence	KDD	Knowledge Discovery in Databases
ANN	Artificial Neural Network	IEEE	Institute of Electrical and Electronic Engineers
ARFF	Attribute Relation File Format	INAF	Istituto Nazionale di Astrofisica
ASCII	American Standard Code for Information Interchange	JPEG	Joint Photographic Experts Group
BoK	Base of Knowledge	LAR	Layered Application Architecture
BP	Back Propagation	MDS	Massive Data Sets
BLL	Business Logic Layer	MLC	Multi Layer Clustering
CE	Cross Entropy	MLP	Multi Layer Perceptron
CSOM	Clustering SOM	MSE	Mean Square Error
CSV	Comma Separated Values	NN	Neural Network
DAL	Data Access Layer	OAC	Osservatorio Astronomico di Capodimonte
DAME	DAta Mining & Exploration	PC	Personal Computer
DAMEWARE	DAME Web Application REsource	PI	Principal Investigator
DAPL	Data Access & Process Layer	REDB	Registry & Database
DL	Data Layer	RIA	Rich Internet Application
DM	Data Mining	SDSS	Sloan Digital Sky Survey
DMM	Data Mining Model	SL	Service Layer
DMS	Data Mining Suite	SOFM	Self Organizing Feature Map
FITS	Flexible Image Transport System	SOM	Self Organizing Map
FL	Frontend Layer	SW	Software
FW	FrameWork	UI	User Interface
GRID	Global Resource Information Database	URI	Uniform Resource Indicator
GSOM	Gated SOM	VO	Virtual Observatory
GUI	Graphical User Interface	XML	eXtensible Markup Language
HW	Hardware		

Tab. 1 – Abbreviations and acronyms

This document contains proprietary information of DAME project Board. All Rights Reserved.

Applicable Documents

ID	Title / Code	Author	Date
A1	SuiteDesign_VONEURAL-PDD-NA-0001-Rel2.0	DAME Working Group	15/10/2008
A2	project_plan_VONEURAL-PLA-NA-0001-Rel2.0	Brescia	19/02/2008
A3	statement_of_work_VONEURAL-SOW-NA-0001-Rel1.0	Brescia	30/05/2007
A4	frontend_VONEURAL-SDD-NA-0004-Rel1.4	Manna	18/03/2009
A5	FW_VONEURAL-SDD-NA-0005-Rel2.0	Fiore	14/04/2010
A6	REDB_VONEURAL-SDD-NA-0006-Rel1.5	Nocella	29/03/2010
A7	driver_VONEURAL-SDD-NA-0007-Rel0.6	d'Angelo	03/06/2009
A8	dm-model_VONEURAL-SDD-NA-0008-Rel2.0	Cavuoti/Di Guido	22/03/2010
A9	dm_model_VONEURAL-SRS-NA-0005-Rel0.4	Cavuoti	05/01/2009
A10	DMPlugins_DAME-TRE-NA-0016-Rel0.3	Di Guido, Brescia	14/04/2010

Tab. 2 – Applicable Documents

This document contains proprietary information of DAME project Board. All Rights Reserved.

__oOo__

This document contains proprietary information of DAME project Board. All Rights Reserved.